ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 6 - Due date 03/25/22

Tatiana Sokolova

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the project open the first thing you will do is change "Student Name" on line 3 with your name. Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A07_Sp22.Rmd"). Submit this pdf using Sakai.

Set up

```
#Load/install required package here
library(forecast)
library(tseries)
library(tidyverse)
library(lubridate)
library(Kendall)
```

Importing and processing the data set

Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

Q1

Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```r
#Importing time series data from text file#
netgen <- read.csv(
  file="../Data/Net_generation_United_States_all_sectors_monthly.csv",
  header=TRUE, skip=4)

#sorting data by Month
netgensort <-
  netgen %>%
  mutate( Month = my(Month) ) %>%
  arrange(Month)

#isolating natural gas
natgas <- data.frame(Month = netgensort$Month, NaturalGas = netgensort$natural.gas.thousand.megawatthou

#Inspecting data
head(natgas)
```

```
##        Month NaturalGas
## 1 2001-01-01   42388.66
## 2 2001-02-01   37966.93
## 3 2001-03-01   44364.41
## 4 2001-04-01   45842.75
## 5 2001-05-01   50934.21
## 6 2001-06-01   57603.15
```

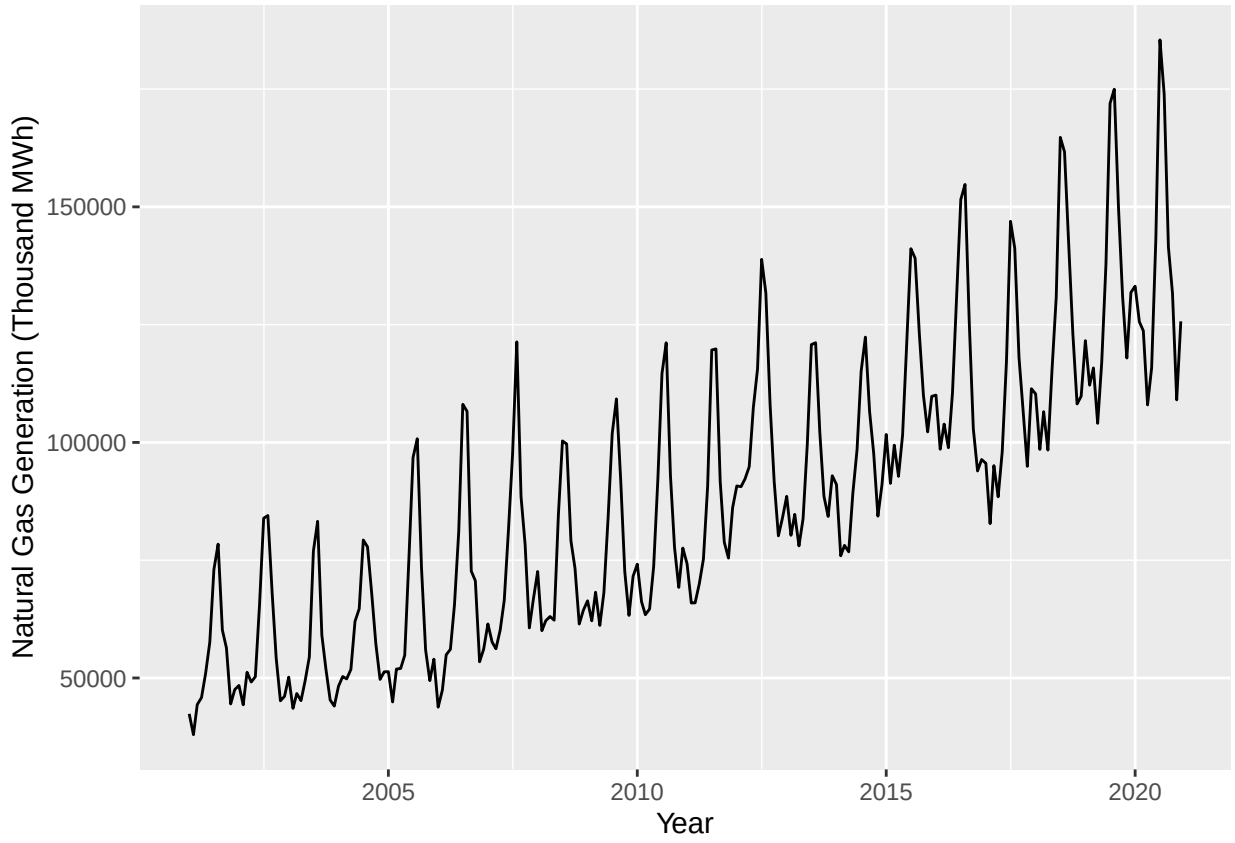
```r
#creating time series
n_for <- 12

ts_natgas <- ts(
  natgas[,2],
  start=c(year(natgas$Month[1]),month(natgas$Month[1])),
  frequency=12)

#sanity check
head(ts_natgas,15)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2001 42388.66 37966.93 44364.41 45842.75 50934.21 57603.15 73030.14 78409.80
## 2002 48412.83 44308.43 51214.46
##           Sep      Oct      Nov      Dec
## 2001 60181.14 56376.44 44490.62 47540.86
## 2002
```

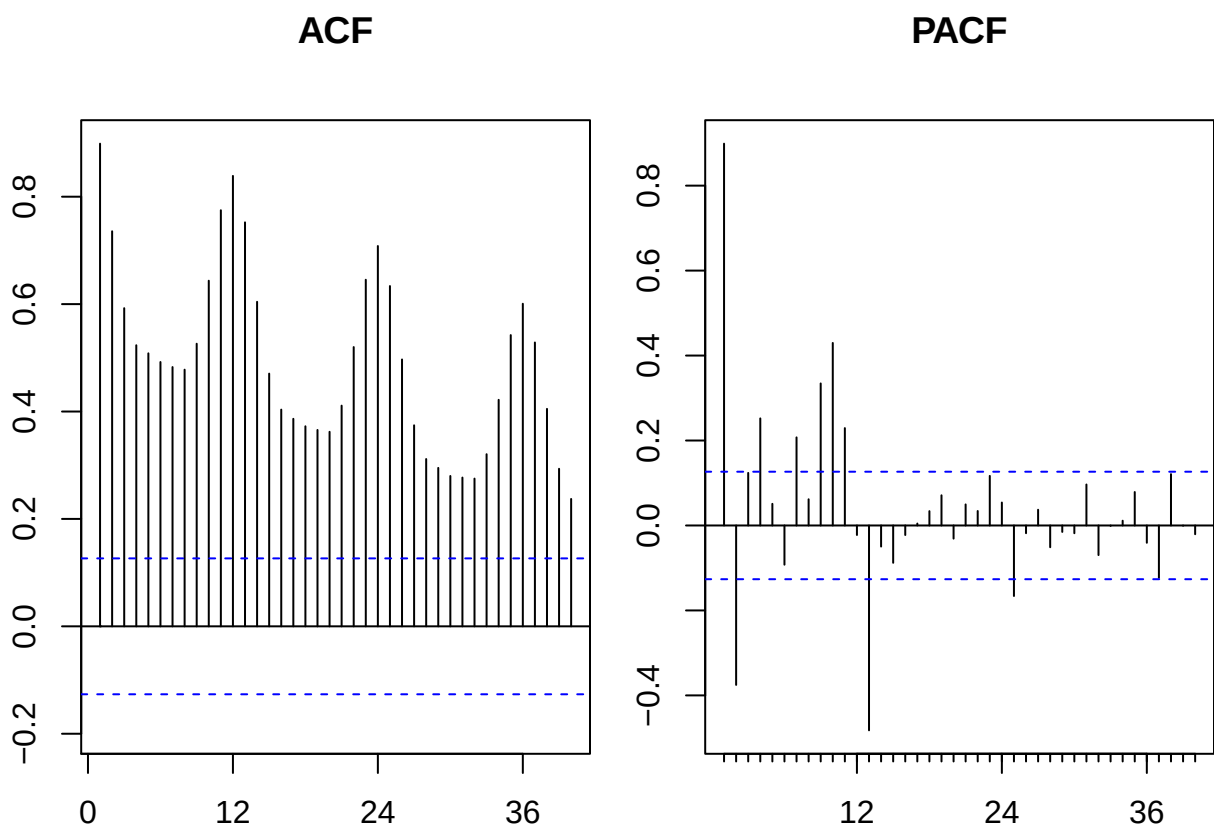
```r
tail(ts_natgas,15)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2019
## 2020 133157.6 125593.9 123697.0 107960.0 115870.9 143245.4 185444.8 173926.6
##           Sep      Oct      Nov      Dec
## 2019          130947.6 117910.5 131838.9
## 2020 141452.7 131658.2 109037.2 125703.7
```

```
#plotting time series
TS_Plot <-
  ggplot(natgas, aes(x=Month, y=NaturalGas)) +
      geom_line() +
      ylab(paste0("Natural Gas Generation (Thousand MWh)")) +
      xlab(paste0("Year"))
plot(TS_Plot)
```



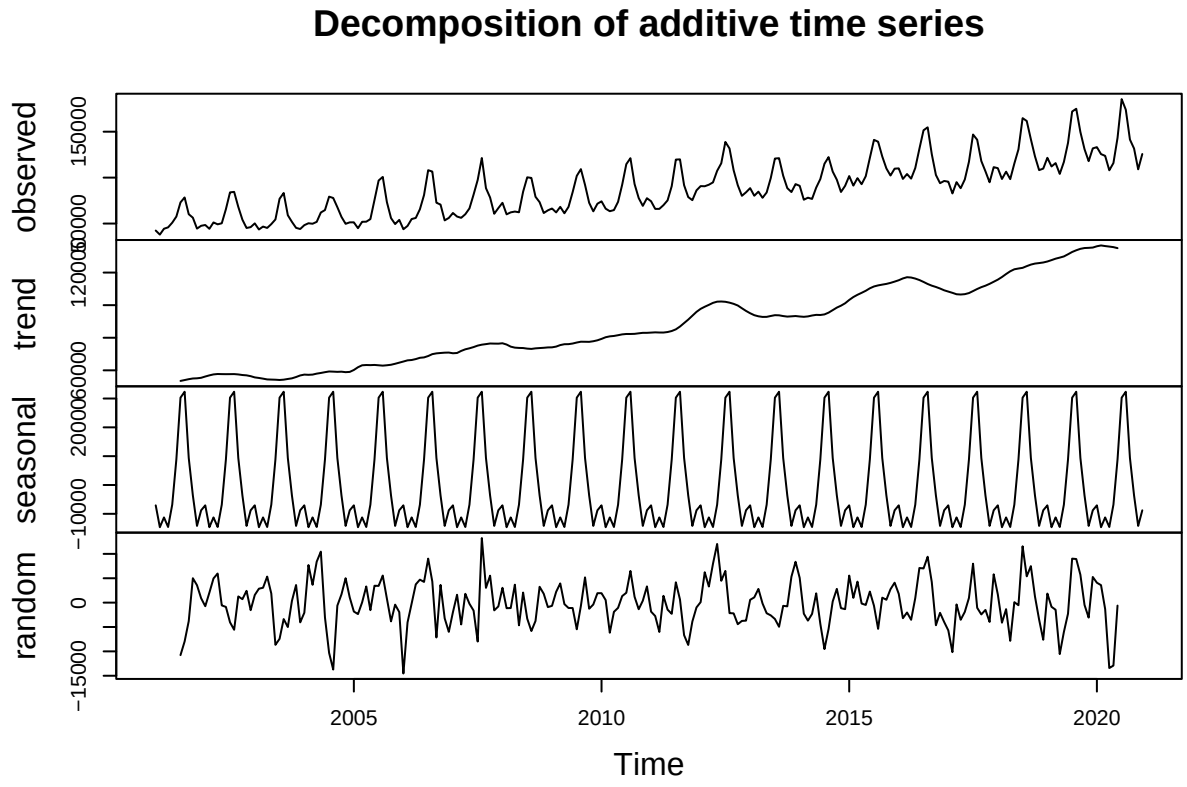

```
#ACF and PACF plots
par(mar=c(3,3,3,0));par(mfrow=c(1,2))
ACF_Plot <- Acf(ts_natgas, lag = 40, plot = TRUE,main="ACF")
PACF_Plot <- Pacf(ts_natgas, lag = 40, plot = TRUE,main="PACF")
```



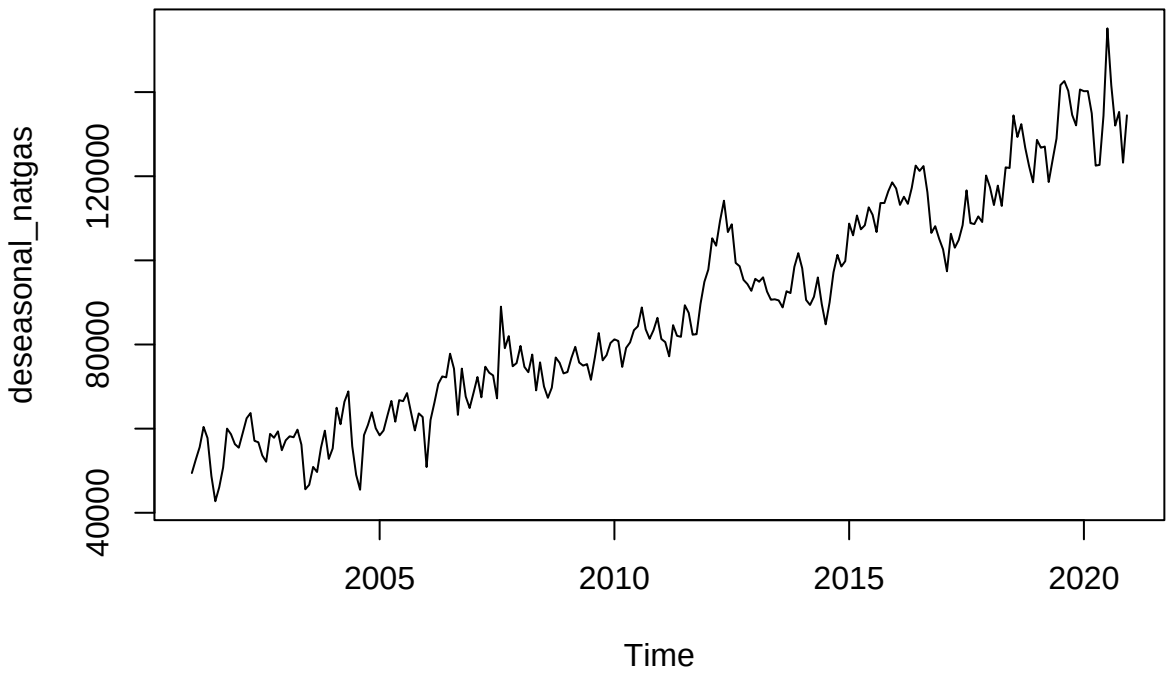
Q2

Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

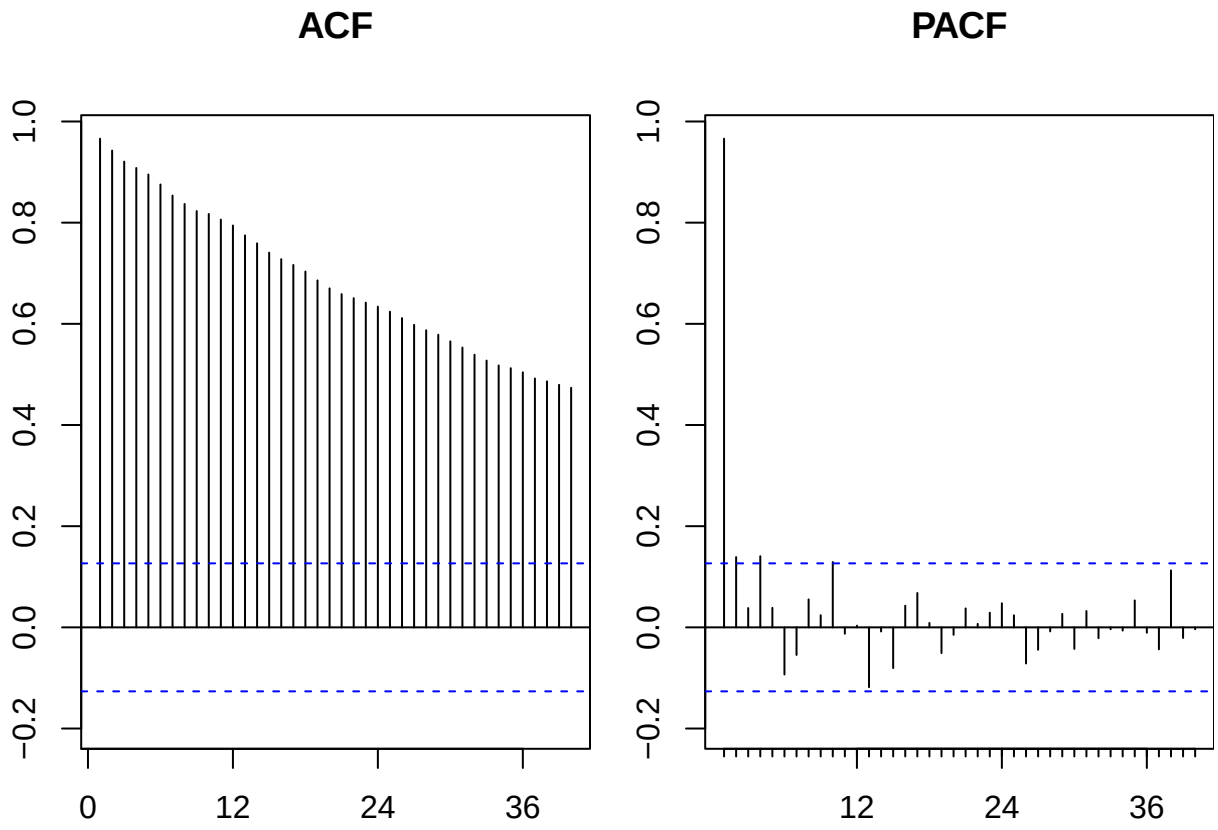
```
#Using decompose function
decompose_natgas <- decompose(ts_natgas,"additive")
plot(decompose_natgas)
```




```
# creating deaseasonalized series
deseasonal_natgas <- seasadj(decompose_natgas)

#plotting deaseasonalized series
plot(deseasonal_natgas)
```




```
#ACF and PACF plots
par(mar=c(3,3,3,0));par(mfrow=c(1,2))
ACF_deseason <- Acf(deseasonal_natgas, lag = 40, plot = TRUE,main="ACF")
PACF_deseason <- Pacf(deseasonal_natgas, lag = 40, plot = TRUE,main="PACF")
```



> Answer: The ACF now is slowly decreasing without the seasonal peaks and valleys. The PACF has a lot more correlation to previous lags than before.

Modeling the seasonally adjusted or deseasonalized series

Q3

Run the ADF test and Mann Kendall test on the deseasonalized data from Q2. Report and explain the results.

```
#ADF Test
print("Results for ADF test")
```

```
## [1] "Results for ADF test"
```

```
print(adf.test(deseasonal_natgas,alternative = "stationary")) #stationary over a unit root but could be
```

```
## Warning in adf.test(deseasonal_natgas, alternative = "stationary"): p-value
## smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_natgas
## Dickey-Fuller = -4.0271, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
#Mann Kendall Test
print("Results of Mann Kendall Test")
```

```
## [1] "Results of Mann Kendall Test"
```

```r
print(summary(MannKendall(deseasonal_natgas)))
```

```
## Score =  24186 , Var(Score) = 1545533
## denominator =  28680
## tau = 0.843, 2-sided pvalue =< 2.22e-16
## NULL
```

Using the ADF test, since the p-value is less than 0.05, we can reject the null hypothesis that the time series has a unit root. E.g. it does not have a unit root. We can therefore conclude the time series is stochastic meaning that it does not have a time-dependent structure.

Using the Mann Kendall test, we can see that the p-value is is statistically different from 0, meaning that the tau value is significant. Therefore, the null hypothesis that there is no trend present in the data should be rejected. E.g. there is a deterministic trend. The larger score indicates that it is a positive trend.

Q4

Using the plots from Q2 and test results from Q3 identify the ARIMA model parameters p, d and q. Note that in this case because you removed the seasonal component prior to identifying the model you don't need to worry about seasonal component. Clearly state your criteria and any additional function in R you might use. DO NOT use the *auto.arima*() function. You will be evaluated on ability to read the plots and interpret the test results.

I will be using ARIMA() from the forecast package and using p=1 because the ACF plot slowly decreases while the PACF has a significant cut off after lag 1, therefore showing an AR model trend. The time series shows no indication of an MA model trend so I will be setting q=0. Since we already have a stochastic trend, I decided that we did not need to difference further and thus chose 0 for d.

Q5

Use *Arima*() from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. Should you allow for constants in the model, i.e., $include.mean = TRUE$ or $include.drift = TRUE$. **Print the coefficients** in your report? Hint: use the *cat*() function to print.

```r
ARIMA_Model<- Arima(deseasonal_natgas,order=c(1,0,0),include.mean = TRUE, include.drift=TRUE)
cat("The AR coefficient is:",ARIMA_Model$coef[1], "and the mean is:", ARIMA_Model$coef[2])
```

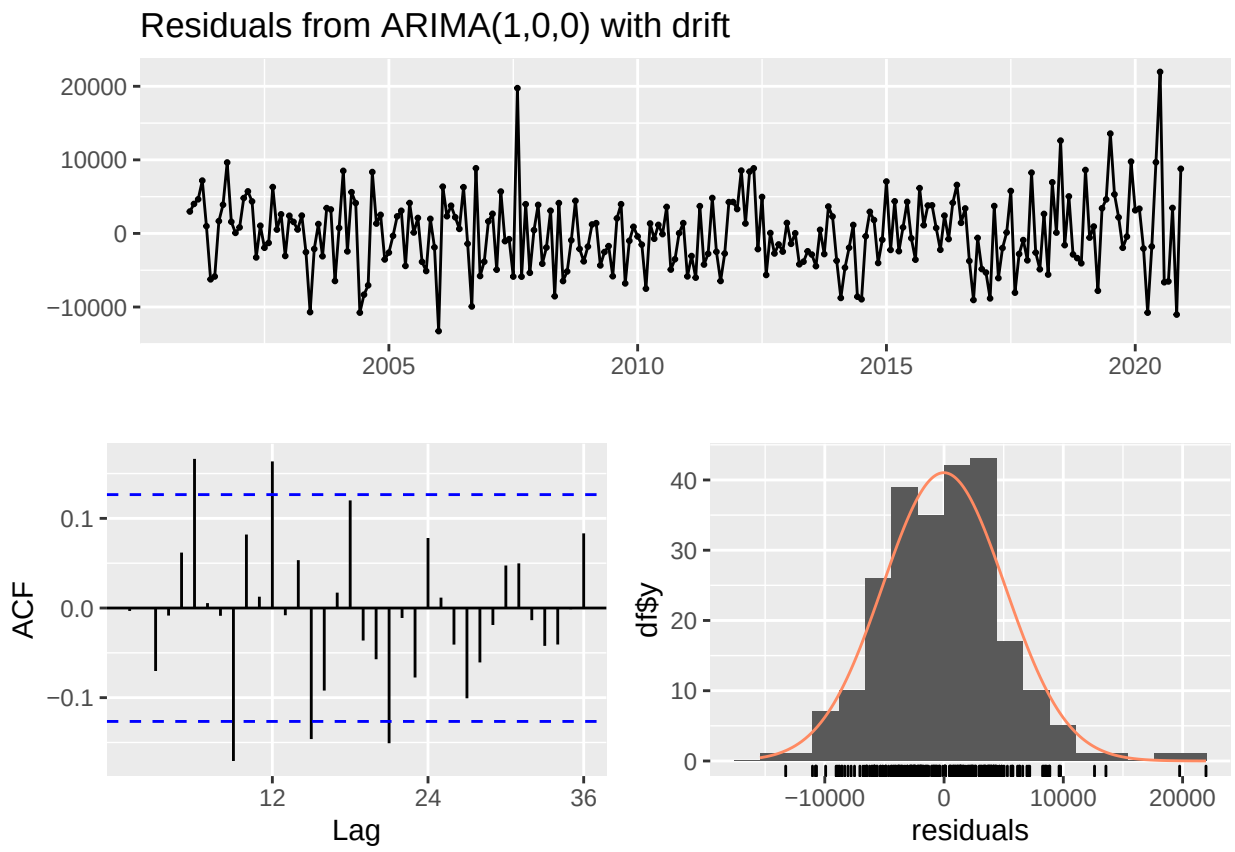
```
## The AR coefficient is: 0.7182166 and the mean is: 44800.49
```

Since I'm not differencing the series, I'm selecting "include.mean = TRUE". However, we generally do not need to use include.mean because it includes the mean by default (I found this out through trial and error). Since we allow drift in most cases, and the series is not differenced twice, I used "include.drift=TRUE" as well.

Q6

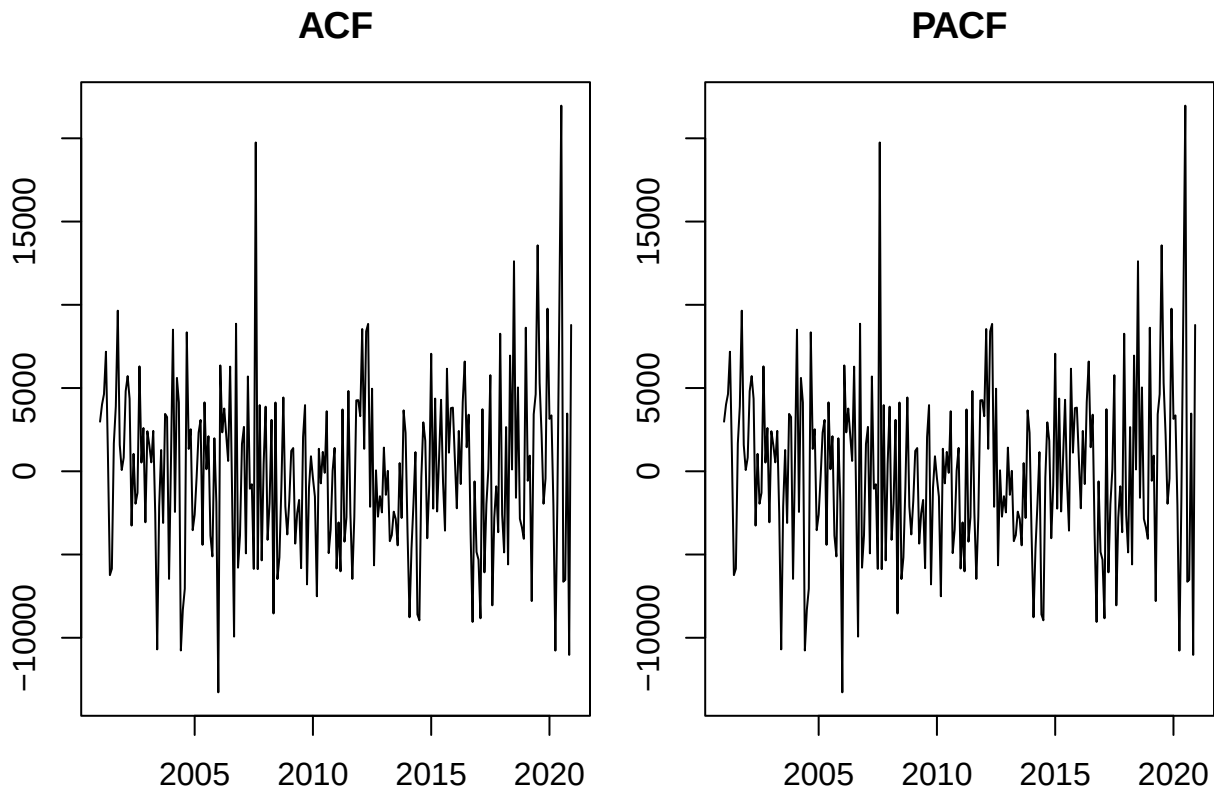
Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```
checkresiduals(ARIMA_Model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with drift
## Q* = 47.775, df = 21, p-value = 0.0007378
##
## Model df: 3.   Total lags used: 24
```

```
par(mar=c(3,3,3,0));par(mfrow=c(1,2))
plot(ARIMA_Model$residuals,main="ACF")
plot(ARIMA_Model$residuals,main="PACF")
```



> The residuals graph looks mostly like a white noise series with some outliers. The ACF appears that it may have some seasonality at each 9th lag but objectively is randomly distributed enough to be considered white noise as well. It follows pretty closely to a normal distribution and therefore also looks pretty close to a white noise series.

Modeling the original series (with seasonality)

Q7

Repeat Q4-Q6 for the original series (the complete series that has the seasonal component). Note that when you model the seasonal series, you need to specify the seasonal part of the ARIMA model as well, i.e., P, D and Q.

> I will be using p=1, d=0, and q=0 as before, along with allowing constants. For the seasonal part of the ARIMA model, since there are multiple spikes in the ACF and one spike in the PACF, I will be using the SAR process, or P=1. Since there is no SMA process trend, I will be using Q=0. Since the seasonal pattern is strong and stable over time, I will be setting D=1 because seasonal differencing is needed.

```
SARIMA_manual <- Arima(ts_natgas,order=c(1,0,0),seasonal=c(1,1,0),include.mean = TRUE, include.drift=TRU
SARIMA_manual
```

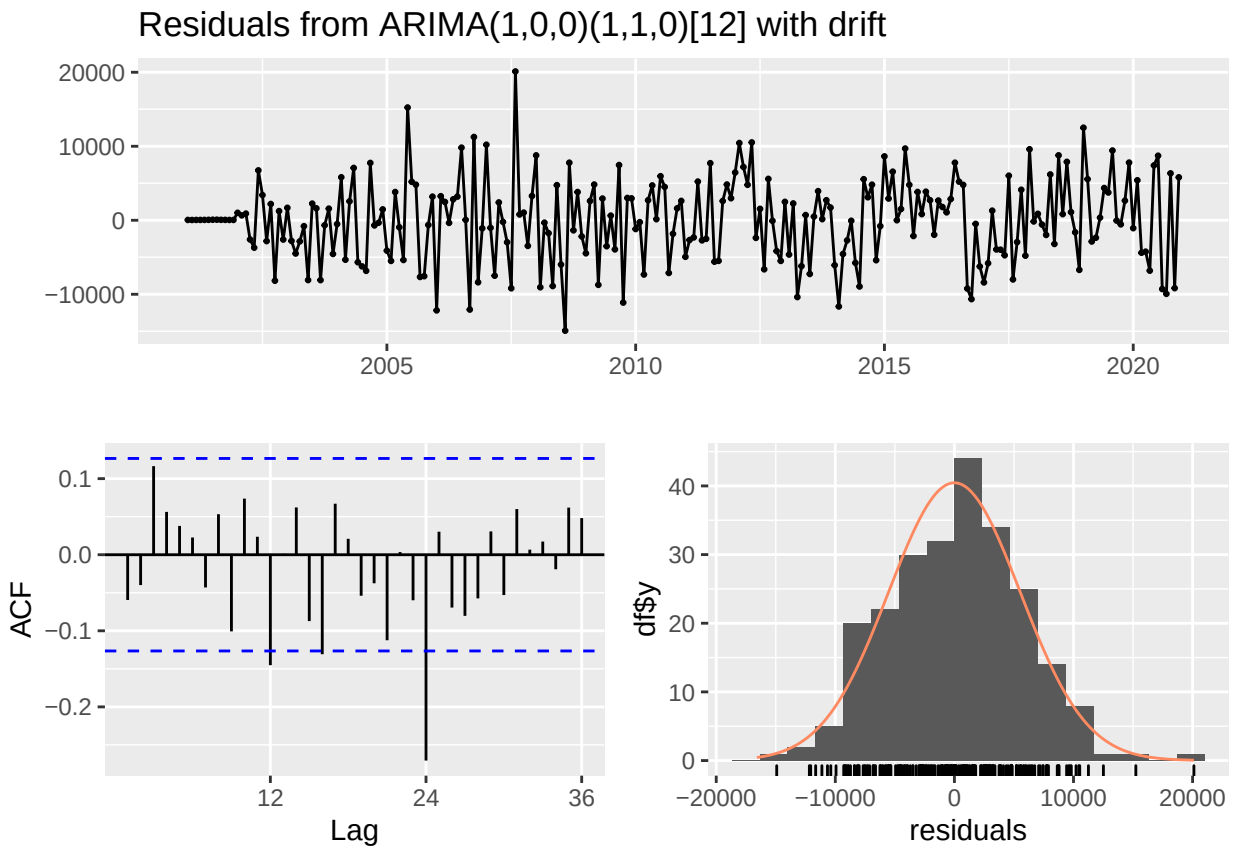
```
## Series: ts_natgas
## ARIMA(1,0,0)(1,1,0)[12] with drift
##
```

```
## Coefficients:
##          ar1     sar1     drift
##       0.7646  -0.4542  358.3892
## s.e.  0.0424   0.0593   91.4518
##
## sigma^2 = 32457520:  log likelihood = -2295.5
## AIC=4598.99   AICc=4599.17   BIC=4612.71
```

```
cat("The AR coefficient is:",SARIMA_manual$coef[1], ", the seasonal MA coefficient is:",SARIMA_manual$c
```

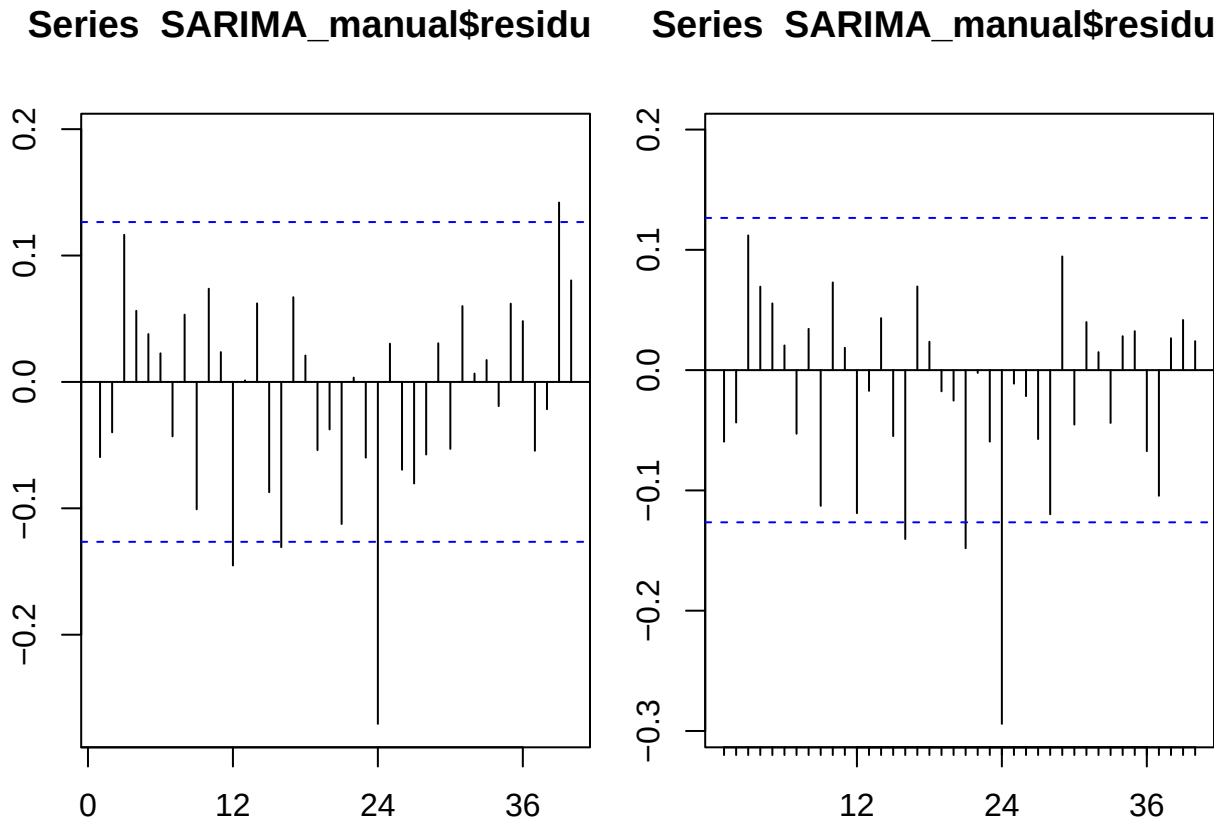
```
## The AR coefficient is: 0.7646127 , the seasonal MA coefficient is: -0.454243 ,and the drift is: 358.3
```

```
checkresiduals(SARIMA_manual)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[12] with drift
## Q* = 50.251, df = 21, p-value = 0.0003366
##
## Model df: 3.   Total lags used: 24
```

```
par(mar=c(3,3,3,0));par(mfrow=c(1,2))
Acf(SARIMA_manual$residuals,lag.max=40)
Pacf(SARIMA_manual$residuals,lag.max=40)
```



Q8

Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

> The SARIMA model better represents the Natural Gas Series because it is more normally distributed and has less outliers. This is not a fair comparison because it is possible that the deaseasoning using the decompose function did not remove all of the seasonality from the time series.

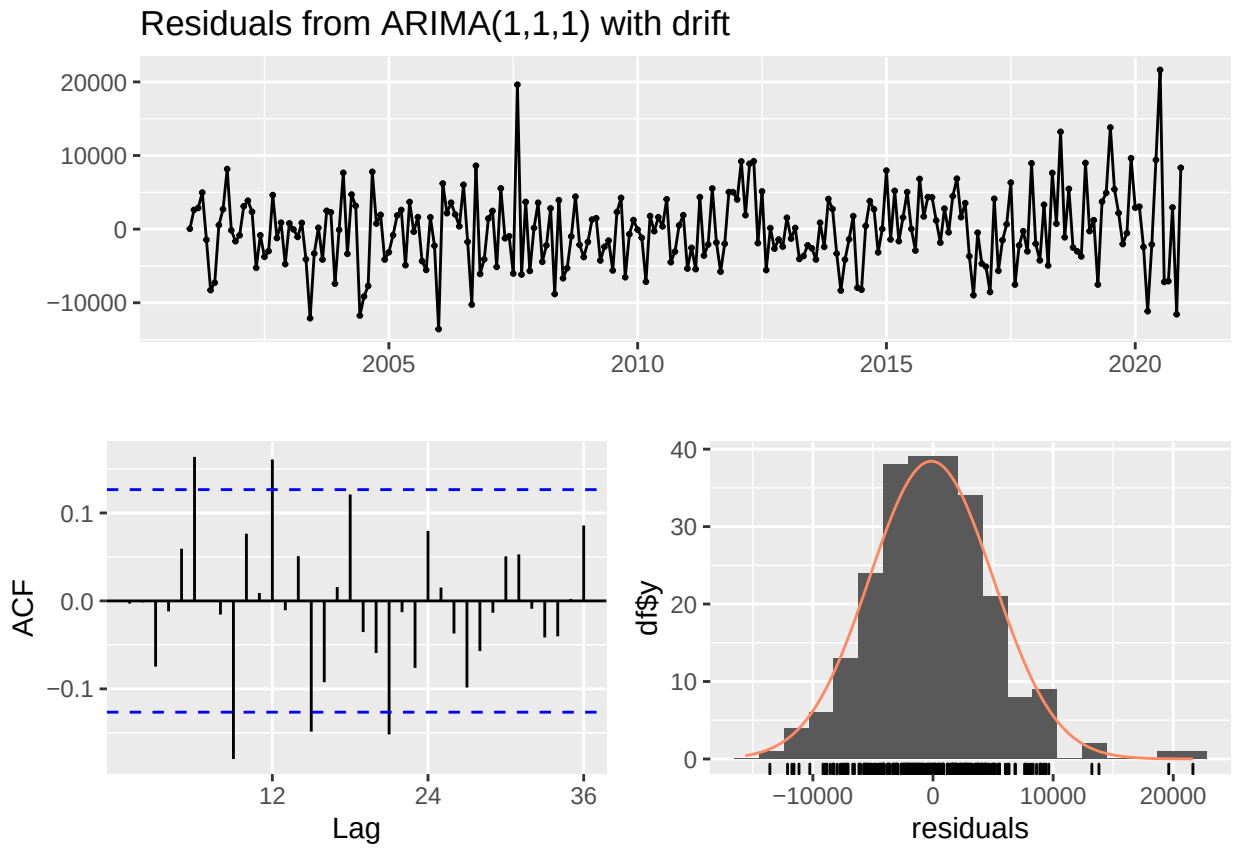
Checking your model with the auto.arima()

Please do not change your answers for Q4 and Q7 after you ran the *auto.arima*(). It is **ok** if you didn't get all orders correctly. You will not loose points for not having the correct orders. The intention of the assignment is to walk you to the process and help you figure out what you did wrong (if you did anything wrong!).

Q9

Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
deaseasoned_autofit <- auto.arima(deseasonal_natgas)
checkresiduals(deaseasoned_autofit)
```



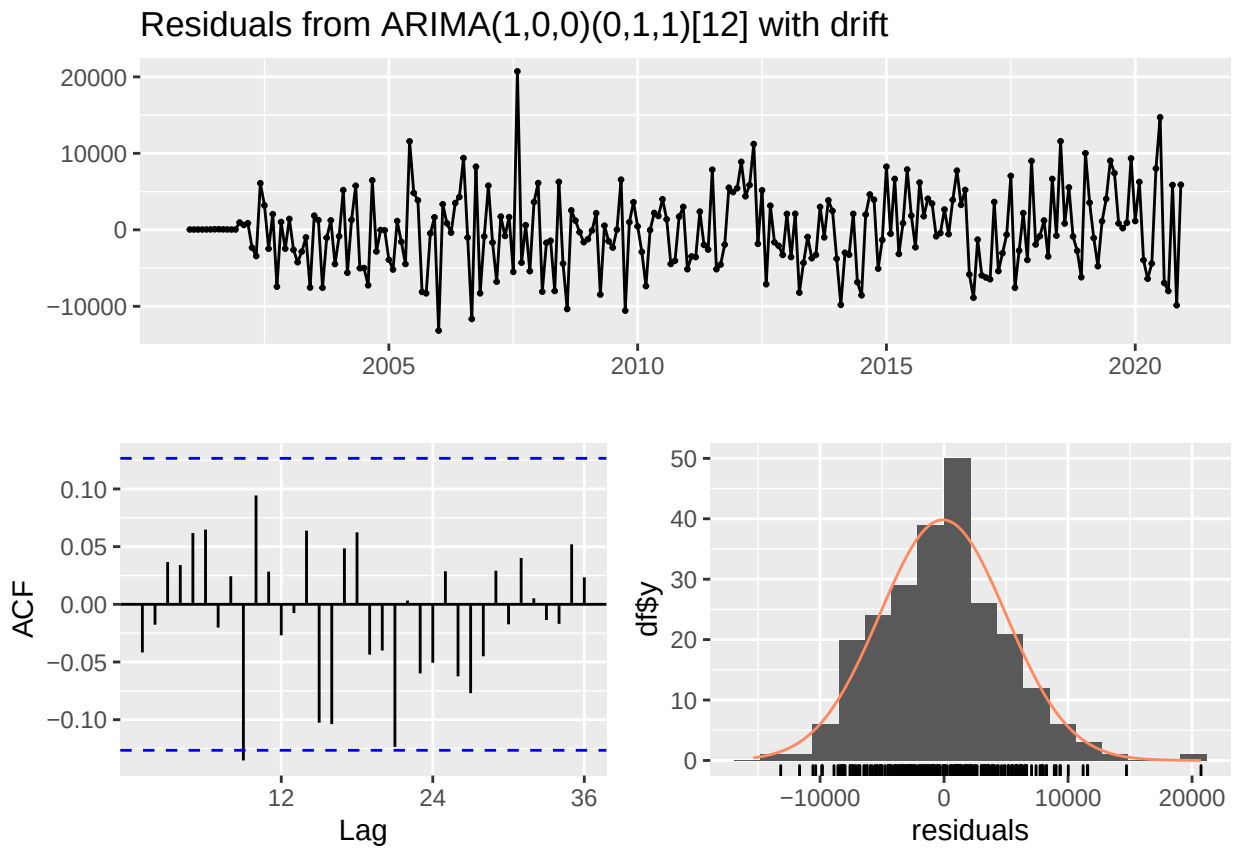

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1) with drift
## Q* = 48.356, df = 21, p-value = 0.000615
##
## Model df: 3.   Total lags used: 24
```

The order of the best model is ARMIA(1,1,1). It does not match what I specified for Q4.

Q10

Use the *auto.arima*() command on the **original series** to let R choose the model parameters for you. Does it match what you specified in Q7?

```
original_autofit <- auto.arima(ts_natgas)
checkresiduals(original_autofit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
## Q* = 25.414, df = 21, p-value = 0.2297
##
## Model df: 3.   Total lags used: 24
```

This does not match what I specified for Q7.